Knowledge-Based Systems 211 (2021) 106525

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A novel framework for detecting social bots with deep neural networks and active learning

Yuhao Wu [a], Yuzhou Fang [a], Shuaikang Shang [b], Jing Jin [b], Lai Wei [a], Haizhou Wang [a,*]

[a] College of Cybersecurity, Sichuan University, Chengdu, 610065, China
[b] College of Computer Science, Sichuan University, Chengdu, 610065, China

ARTICLE INFO

Article history:
Received 28 May 2020
Received in revised form 26 September 2020
Accepted 12 October 2020
Available online 22 October 2020

Keywords:
Online social networks
Social bots
Sina Weibo
Deep neural networks
Active learning

ABSTRACT

Microblogging is a popular online social network (OSN), which facilitates users to obtain and share news and information. Nevertheless, it is filled with a huge number of social bots that significantly disrupt the normal order of OSNs. Sina Weibo, one of the most popular Chinese OSNs in the world, is also seriously affected by social bots. With the growing development of social bots in Sina Weibo, they are increasingly indistinguishable from normal users, which presents more huge challenges in detecting social bots. Firstly, it is difficult to extract the features of social bots completely. Secondly, large-scale data collection and labeling of user data are extremely hard. Thirdly, the performance of classical classification approaches applied to social bot detection is not good enough. Therefore, this paper proposes a novel framework for detecting social bots in Sina Weibo based on deep neural networks and active learning (DABot). Specifically, 30 features from four categories, namely metadata-based, interaction-based, content-based, and timing-based are extracted to distinguish between social bots and normal users. Nine of these features are completely new features proposed in this paper. Moreover, active learning is employed to efficiently expand the labeled data. Then, a new deep neural network model called RGA is built to implement the detection of social bots, which makes use of a residual network (ResNet), a bidirectional gated recurrent unit (BiGRU), and an attention mechanism. After performance evaluation, the results show that DABot is more effective than the state-of-the-art baselines with the accuracy of 0.9887.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

In the past decades, online social networks (OSNs) have enabled users to conduct massive-scale and real-time communication and have had a significant impact on public life [1]. OSNs provide convenience for users to keep in touch with family members, friends, etc. Moreover, it is very easy for users to get the latest news from OSNs [2]. While being widely used, OSNs have gradually emerged a new class of program-controlled users, namely social bots. At first, these social bots, including bots that automatically aggregate content from various sources and bots for replying to inquiries [3] were used to serve users. However, the rise of malicious social bots has caused harm to OSNs and people in the real world [4]. Malicious social bots are users with illicit purposes controlled by programs in OSNs. As malicious social bots continue to evolve, their behaviors including the guidance of online public opinion, malicious commentary, defamation, and ideology infiltration, have posed huge damage to normal social order and even national stability.

As one of the largest Chinese microblogging services in the world, Sina Weibo has a significant influence on the Chinese social community. With the gradual popularity of Sina Weibo, it has also become one of the most active OSNs of malicious social bots. Thus, detecting and filtering social bots in Sina Weibo is of great importance.

1.1. Social bots in OSNs

Nowadays, social bots are used for three main purposes in OSNs. One class of social bots, known as social spammers (also called Internet water armies in some cases) is used for obtaining benefits through illegal use [5]. They can post a lot of advertisements intensively, spread malicious URLs, and publish rumors to mislead other users [6]. Another class of social bots is generally used to increase the popularity of target users by following them. These social bots also repost, like, and comment on specific posts [7]. The third class of social bots can be used to interfere with political activities and guide public opinion. For example, during the presidential election in the U.S. in 2016, social bots spread a large number of fake tweets on Twitter, and the decisions of many voters were affected by such tweets [8].

* Corresponding author.
E-mail address: whzh.nc@scu.edu.cn (H. Wang).

https://doi.org/10.1016/j.knosys.2020.106525
0950-7051/© 2020 Elsevier B.V. All rights reserved.

A lot of research has been done on social bot detection in OSNs [9–16]. However, most of the existing work of social bot detection is launched on Twitter [9–11] and Facebook [12–14], and there are relatively few studies based on the Chinese OSNs, such as Sina Weibo [15,16]. Because of the differences in languages, functions, and features of social bots in different OSNs, it is difficult to directly migrate detection technologies based on other OSNs to Sina Weibo. Therefore, the research on social bot detection in Sina Weibo needs to be undertaken more comprehensive and in-depth.

1.2. Challenges

At present, research on social bot detection in Sina Weibo mainly faces the following three challenges:

The first challenge is that the features of social bots in Sina Weibo are complex, and it is difficult to extract features completely. Social bots often pretend to be normal users to avoid being detected. It is necessary to consider the features of social bots from multiple aspects so as to describe them more accurately. Many existing studies only extract the features of social bots from a single perspective [16,17] and cannot describe them completely. Besides, some work just uses a small number of features to build a detection model, although the features of social bots from a couple of aspects are considered [15,18].

The second challenge is that it is difficult to obtain a large-scale labeled dataset for research from Sina Weibo. Due to the relatively rare social bot detection research on Sina Weibo, there is a lack of large-scale reliable datasets. In the meantime, labeling samples manually requires rich experience support, and a lot of time cost. Most of the existing research is based on small-scale datasets [16,18,19]. Therefore, how to build a large-scale dataset accurately and efficiently is another great challenge in current research on social bot detection in Sina Weibo.

The third challenge is that the classical detection approaches do not perform quite well in detecting social bots in Sina Weibo. Although some machine learning detection approaches have been used by previous work [16,20], there is still a lot of work to do to improve the performance of the detection approaches. Hence, a high-performance social bot detection approach based on deep neural networks needs further development.

1.3. Contribution and organization

As for the above challenges, this paper proposes a novel framework, i.e. DABot, which takes advantages of **D**eep neural networks and **A**ctive learning for social **Bot** detection in Sina Weibo. DABot consists of four modules: data collection and labeling module, feature extraction module, active learning module, and detection module. To begin with, the data collection and labeling module is responsible for collecting user data from Sina Weibo, and then manually labeling a small set of collected data. Next, the feature extraction module is used to analyze and extract the features of social bots and normal users. Furthermore, the active learning module expands the labeled data through the active learning approach [21]. Eventually, the RGA model, which is based on a **R**esidual network (ResNet) [22], a bidirectional **G**ated recurrent unit (BiGRU) [23], and an **A**ttention mechanism [24] is designed to detect social bots in the detection module. The main contributions of this paper are summarized as follows:

- A complete framework DABot for detecting social bots in Sina Weibo is proposed, which mainly integrates deep neural networks and active learning, and it achieves excellent detection performance.
- A total of 30 features are extracted to identify social bots in Sina Weibo accurately, and nine of them are completely new features. All the features can be divided into four categories: metadata-based, interaction-based, content-based, and timing-based features.
- An active learning-based labeled data expansion approach is proposed, and a large-scale and balanced dataset with 300,000 samples is constructed.
- A novel deep neural network called RGA is built for detection, which takes advantage of the ResNet, the BiGRU, and the attention mechanism. The evaluation results show that it significantly outperforms the widely used detection approaches.

The rest of this paper is organized as follows. In Section 2, related work and achievements in the field of social bot detection are introduced. The proposed framework DABot is elaborated in Section 3. Furthermore, Section 4 describes the experimental setup and evaluation results. finally, Section 5 concludes the research and plans for future work.

2. Related work

In this section, we summarize the studies on social bot detection in OSNs in recent years. The approaches in social bot detection studies fall into three main categories: graph-based approaches, machine learning approaches, and other approaches. The related work of each category of detection approach is introduced separately below.

2.1. Graph-based approaches

In the study of graph-based detection approaches, social bots are often referred to as sybils (fake accounts) [25]. These social bots are used to create multiple identities to destroy reputation systems and carry out other malicious attack activities. The graph-based approach is mainly to establish a social network graph for detection based on the relationships and behaviors among users. However, social bots can evade them by creating sufficient attack links (edges) between normal users and themselves. Some graph-based detection approaches are introduced as follows.

In [26], a semi-supervised learning framework called SybilBelief was proposed to detect sybil nodes. SybilBelief takes a social network of the nodes (a small set of known benign nodes, and a small set of known sybils) in the system as input. Then, SybilBelief propagates the label information from the nodes of known benign or sybil to the remaining nodes in the system. However, the number of accepted sybil nodes increases dramatically when the labeled benign and sybil nodes are highly imbalanced.

Over time, Yang et al. [27] proposed a scalable defense system called VoteTrust. VoteTrust models the invitation interactions of friends among users as a directed and signed graph. Two key mechanisms are used to detect sybils over the graph: a voting-based sybil detection to find sybils that users vote to reject, and a sybil community detection to find other colluding sybils around identified sybils. During the same period, Boshmaf et al. [13] designed Íntegro, a scalable defense system that uses a robust scheme of user ranking. Íntegro starts by predicting victim users from user-level activities. Then, these predictions are integrated into the graph as weights. finally, Íntegro ranks users based on a modified random walk that starts from a real user. Íntegro achieves that the ranks of most real users are higher than fake accounts.

In [28], a structure-based approach called SybilSCAR was proposed. SybilSCAR unifies random walk (RW)-based and loop belief

propagation (LBP)-based approaches, which is scalable, convergent, accurate, and robust to label noises. To improve the robustness of the sybil detection, a two-layer hyper-graph model called SybilSAN was proposed by Zhang et al. [29], which fully uses users' friendships and their corresponding activities in OSNs. Markov chain mixing time is employed to derive the number of rounds needed to guarantee that the iterative algorithm terminates. Moreover, the graph is divided into three sub-graphs, and a random walk is designed to propagate trust independently for each sub-graph. finally, SybilSAN uses a unified algorithm to couple these three random walks to capture a mutual relationship between users and activities.

2.2. Machine learning approaches

Currently, machine learning-based approaches are the most widely used approaches in the field of social bot detection. The machine learning approaches can be categorized into classical machine learning approaches and deep learning approaches, which are introduced separately as below.

2.2.1. Classical machine learning approaches

The classical machine learning approach performs social bot detection in OSNs by training a classical machine learning classifier.

Chu et al. [30] studied the automation by bots and cyborgs in Twitter. To better understand the role of automation on Twitter, they measured and characterized the behaviors of humans, bots, and cyborgs on Twitter and proposed new features. Also, an automated classification system was designed for detecting social bots. After that, Yang et al. [31] made an empirical analysis of the evasion tactics utilized by Twitter spammers and further designed several new detection features to detect Twitter spammers. In their work, a large-scale dataset was constructed, and random forest (RF), decision tree (DT), and some other classical machine learning classifiers were applied. finally, RF achieved the best performance with the F1-score reaching 0.9000.

In [32], Miller et al. modified two stream clustering algorithms, StreamKM++ and DenStream, to facilitate social bots identification in Twitter. Also, 95 one-gram features from tweet text were introduced alongside the user information. finally, each of these algorithms performed well individually, with StreamKM++ achieving a 0.0640 false positive rate and DenStream reaching a 0.0280 false positive rate. Cai et al. [33] proposed an extreme learning machine (ELM)-based approach for effectively detecting social bots in Sina Weibo. They first constructed the dataset through crawling user data from Sina Weibo and manually labeled data. Then, features were extracted from message content and behavior, and the ELM is applied to detect spammers.

Al-Qurishi et al. [9] proposed an integrated social media content analysis platform that leverages three levels of features, user-generated content, social graph connections, and user profile activities to detect social bots in Twitter and YouTube. In addition, they proposed novel approaches in the process of data extraction and classification to contextualize the large-scale networks. Supervised machine learning classifiers such as support vector machine (SVM), RF, etc. were applied to detect social bots, and RF got the highest accuracy with 0.9607.

2.2.2. Deep learning approaches

Deep learning achieves excellent results in image classification [34], recommender systems, document clustering [35], etc. Also, in the past few years, deep learning approaches begin to be more wildly used in social bot detection. Comparing with classical machine learning approaches, deep learning approaches have better generalization performance and are more suitable for processing big data.

In [11], a social bot detection approach based on deep learning algorithm was proposed, i.e. DeBD, which reached an average accuracy with 0.9760. In [10], a deep neural network based on contextual long short-term memory (LSTM) architecture that exploits both content and metadata to detect social bots was proposed. The authors also proposed a technique based on synthetic minority oversampling to generate a large-scale labeled dataset. The results showed that their approach had a high area under the curve (AUC) in detecting social bots.

Recently, reinforcement learning has been used in this field. Lingam et al. [7] designed a deep Q-network architecture by incorporating a deep Q-learning (DQL) model using the social attributes for detection of social bots based on updating Q-value function. In their work, tweet-based features, user profile-based features, and social graph-based features of users were extracted respectively. The experimental results showed that the DQL algorithm provides 5%–9% improvement of precision over the baseline algorithms.

2.3. Other approaches

In addition to the graph-based approaches and the machine learning approaches, there are some other approaches for detecting social bots such as the detection approaches for social bot clusters with joint attack properties. A brief introduction of some other approaches is as follows.

In [36], Chavoshi et al. proposed a correlation approach based on dynamic time warping (DTW) and developed a novel lag-sensitive hashing technique for discovering social bots with highly time-dependent activities. The paired DTW distances calculated from the time series are clustered through the approach. After that, based on the similarity of users, Cresci et al. [37] used digital DNA technology in this field and achieve the efficient detection of social bots.

Unlike the above approaches, Zhao et al. [38] proposed an approach that comprehensively considers all target users' decisions about finding the best actions against social bots. Each target user can investigate the average rating of each source user. If the average rating is lower than a threshold, the target user thinks that the source user is more likely to be a social bot. In [39], an interactive and visual social bot annotation system called VASSL was designed to improve the efficiency of labeling samples, which significantly improved the efficiency of detecting social bots.

3. The proposed framework for detecting social bots

In this section, we describe in detail the proposed framework DABot for detecting social bots in Sina Weibo, which is shown in Fig. 1. DABot is composed of a data collection and labeling module, a feature extraction module, an active learning module, and a detection module.

(1) **Data collection and labeling module:** This module is responsible for collecting user data from Sina Weibo and manually labeling a small portion of user data, which provides effective data support for other modules. In this module, a web crawler to collect data from Sina Weibo is developed, and six discrimination metrics for manually labeling is proposed. A small set of the collected data is manually labeled based on the metrics.

(2) **Feature extraction module:** The main task of this module is to analyze and extract the features of social bots and normal users in Sina Weibo, so as to construct feature vectors of users. In this module, 30 features of users

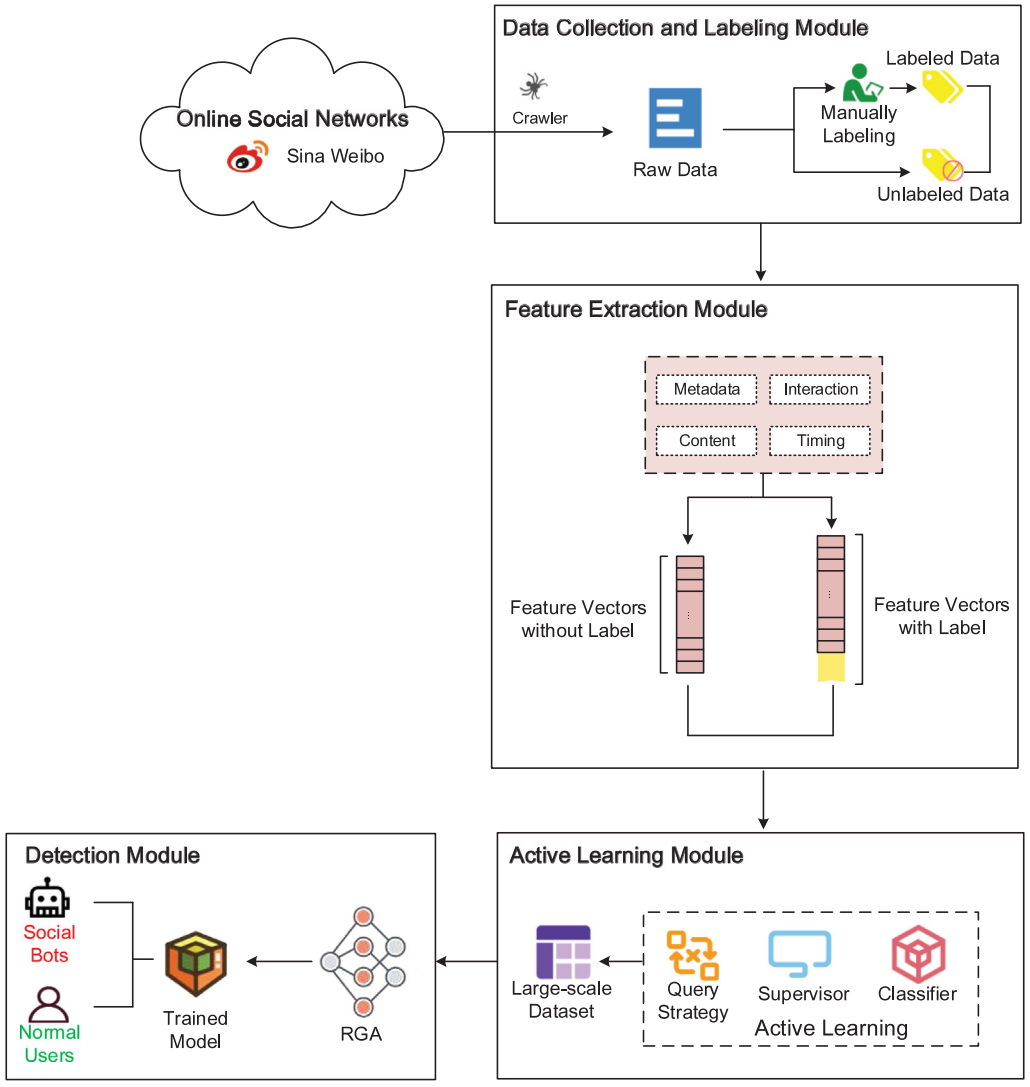

Fig. 1. The architecture of the proposed social bot detection framework DABot.

are extracted to identify social bots. These features are divided into four categories, including metadata-based, interactive-based, content-based, and timing-based features. This module is the basis for the construction of experimental datasets.

(3) **Active learning module:** This module is mainly responsible for building a large-scale experimental dataset using active learning. It is composed of a query algorithm, a machine learning classifier, and a supervisor. It can efficiently expand the labeled data and facilitate the training of subsequent detection models.

(4) **Detection module:** This module implements the detection of social bots. It adopts the detection approach based on a novel deep neural network, which makes use of the ResNet, the BiGRU, and the attention mechanism. The detection approach can completely utilize the feature vectors of users to achieve much more accurate detection performance.

The details of each module of the proposed framework DABot are described below.

3.1. Data collection and manually labeling

At present, there is relatively little research on social bot detection in Sina Weibo, resulting in a lack of reliable datasets of Sina Weibo comprising social bots and normal users. Therefore, we develop a web crawler of Sina Weibo and collect a large amount of user data from Sina Weibo. Furthermore, we manually label part of user data and construct a dataset with a sample size of 20,000.

3.1.1. Data collection approach

Developer APIs for data accessing are provided by Sina Weibo, which are great ways for researchers and developers to collect user data from Sina Weibo at no cost. Nevertheless, there are some strict restrictions on data collection using these APIs. To meet the needs of the study, a high-performance multi-threaded web crawler is developed, by which multi-tasks with multiple proxy IP can be created to cycle through them and a series of API requests can be built to download raw HTML data from the web. After that, the valid data such as user profiles and posts of users are extracted, which is stored in a database next.

3.1.2. Data collection of potential normal users and potential social bots

In order to collect user data of Sina Weibo more efficiently, we analyze the characteristics of the distribution and the behavior of normal users and social bots firstly. Then, we design collection strategies for potential users and complete user data collection.

Table 1
Overview of the collection of collected user data *SWRD*.

Category	User number	Post number
Potential normal users	228,768	8,863,520
Potential social bots	183,591	6,033,711
Total	412,359	14,897,231

Table 2
Overview of the $SWLD-20K$ dataset.

Category	User number	Post number
Normal users	10,000	118,199
Social bots	10,000	96,307
Total	20,000	214,506

(1) **Data collection for potential normal users:** There is a service in Sina Weibo called *local service* that pushes other local active users to the current user. After further research on the users pushed by *local service*, we find that most of these users have many original posts and interact frequently with others. Since these users have passed the screening and filtering of Sina Weibo, most of these users are normal users. Therefore, these users are regarded as potential normal users, and we collect the user data of them by the web crawler.
(2) **Data collection for potential social bots:** In general, it is difficult to correctly identify the social bots from a large number of users with absolute accuracy, unless the social bot promotion companies provide real data. Therefore, in order to obtain the most accurate social bot samples at the beginning, we purchase 5000 social bots from five social bot promotion companies. By observing these social bots, we find that few normal users follow social bots, instead, social bots usually follow each other to increase their own influence. Consequently, most of the users in the follower list of existing social bots are social bots. Further, these users are considered as potential social bots and their user data is collected.

Using the data collection approach with the collection strategies for potential normal users and social bots, we collect user data of a total of 412,359 users eventually, which is represented as *SWRD* (Sina Weibo raw data), as shown in Table 1. Note that the collection time of the user data is from October 25, 2019 to January 24, 2020.

3.1.3. Manually labeling

After completing the data collection of potential social bots and potential normal users, we only manually label a small set of user data, because manually labeling is very time-consuming. According to the differences between normal users and social bots in the profile information, content of posts, and dynamic behavior characteristics [30], we propose six metrics to distinguish them during manually labeling, which are as follows:

(1) **The integrity of user profile:** The profile of normal users in Sina Weibo is generally complete, while the profile of social bots is usually missing information.
(2) **The rationality of the users' social relationship:** Since social bots are widely used to increase the number of followers for others, they usually follow more users than normal users, making their social relationships less reasonable than normal users.
(3) **Frequency of interaction with other users:** Since most of the normal users have their own circle of friends, their posts generally have more likes, comments, and reposts, while social bots have lower interaction frequency due to the lack of normal users' attention. Although social bots are sometimes used to like, comment, and repost other users' posts, their own posts lack influence.
(4) **The originality of users' posts:** In Sina Weibo, social bots are often used for malicious reposting in order to guide public opinion. Therefore, as for social bots, the proportion of the number of reposts to the total number of posts is much larger than normal users.
(5) **Regularity of the time of posting:** In order to guide public opinions in some hot events, such as political election, many social bots post a lot of posts in a very short time, while normal users do not.
(6) **Quality of original post content:** The posts of normal users are usually more logical and complete in expression. However, the posts of social bots often have more problems such as the misuse of punctuation, unclear semantics, and confusion of context logic.

According to these metrics, we firstly select some potential normal users and potential social bots to manually label them. Subsequently, we extract the features of all users in *SWRD* according to the feature extraction approach described in Section 3.2. finally, we construct a labeled dataset called $SWLD-20K$ (Sina Weibo labeled dataset with 20,000 samples), which contains data of 10,000 normal users and 10,000 social bots. The description of the $SWLD-20K$ dataset is shown in Table 2. The remaining data of potential normal users and potential social bots is formed into an unlabeled dataset *SWUD* (Sina Weibo unlabeled dataset).

3.2. Feature analysis and extraction

In our work, we analyze and extract features of social bots and normal users, which are divided into four categories: metadata-based features, interaction-based, content-based as well as timing-based features. A total of 30 features are extracted, of which nine features are completely new features proposed in this paper. Table 3 briefly summarizes the features, where * represents that the feature has been defined in existing research work, but we redefine it.

3.2.1. Metadata-based features

Metadata-based features are extracted from users' profile which includes the nickname, the number of followers, the number of following, the introduction, the location, etc. These data can reveal the differences between normal users and social bots. Based on these data, we propose the following six metadata-based features.

(1) **Length of nickname:** The length of nickname is used as a feature and achieved good results in [40]. Therefore, we also adopt the length of nickname as a feature and represent it as β_{LN}. Note that Sina Weibo has strict restrictions on the length of the nickname, the value range of β_{LN} is $\{\beta_{LN}|2 \le \beta_{LN} \le 30\}$.
(2) **Ratio of followers to following:** The ratio of followers to following is considered in [2,15]. In our work, α denotes the number of followers of a user and ς denotes the number of following. We represent the ratio of followers to following as β_{RFF}, which is given by

$$\beta_{RFF} = \frac{\alpha}{\alpha + \varsigma}. \tag{1}$$

The value range of β_{RFF} is $\left\{\beta_{RFF}\middle|0 \le \beta_{RFF} \le 1\right\}$..
(3) **Default nickname and avatar:** Lots of social bots use the default nicknames and avatars [40]. Hence, whether a user uses the default nickname and whether a user uses the

Table 3
Summary of all the extracted features.

Category	Feature name	Symbol	Source
Metadata-based	Length of nickname	β_{LN}	[40]
	Ratio of follower to following	β_{RFF}	[2,15]
	Default nickname	β_{DN}	[40]
	Default avatar	β_{DA}	[40]
	Completeness of profile	β_{CP}	[31]
	Comprehensive level	β_{CL}	Our work
Interaction-based	Mean of the number of comments of posts	γ_{MNCP}	[41]
	Mean of the number of reposts of posts	γ_{MNRP}	[41]
	Mean of the number of likes of posts	γ_{MNLP}	[41]
	Diversity of sources of posts	γ_{DSP}	Our work
	Repost ratio	γ_{RR}	[2,42]
Content-based	Mean of the number of mentions in posts	δ_{MNMP}	[2]
	Variance of the number of mentions in posts	δ_{VNMP}	[2]*
	Mean of the number of hashtags in posts	δ_{MNHP}	[2,30]
	Variance of the number of hashtags in posts	δ_{VNHP}	[2,30]*
	Mean of the number of URLs in posts	δ_{MNUP}	[2,30]
	Variance of the number of URLs in posts	δ_{VNUP}	[2,30]*
	Variance of the number of words in posts	δ_{VNWP}	[40]*
	Mean of the number of punctuation marks in posts	δ_{MNPMP}	Our work
	Variance of the number of punctuation marks in posts	δ_{VNPMP}	Our work
	Mean of the number of interjections in posts	δ_{MNIP}	Our work
	Variance of the number of interjections in posts	δ_{VNIP}	Our work
	Mean of the score of the sentiment of posts	δ_{MSSP}	[40,43]
	Variance of the number of pictures in posts	δ_{VNPP}	Our work
Timing-based	Mean of the time interval between posts	φ_{MTIP}	[15]*
	Variance of the time interval between posts	φ_{VTIP}	[15]*
	Shortest time interval between posts	φ_{STIP}	Our work
	Longest time interval between posts	φ_{LTIP}	Our work
	Burstiness parameters of the time interval between posts	φ_{BPTIP}	[17]
	Information entropy of the time interval between posts	φ_{IETIP}	[17]

default avatar are respectively represented as β_{DN} and β_{DA} in our work. The value of β_{DN} is 1 if the user uses the default nickname, otherwise, it is 0. And the calculation of β_{DA} is similar to β_{DN}.

(4) **Completeness of profile:** Users of Sina Weibo can fill in or change their profile. Normal users have real friend-making demands, so they usually fill in their profiles carefully. However, the profile of social bots is usually incomplete [31]. We use the completeness of profile as a feature, and it is given by

$$\beta_{CP} = \sum_{i=1}^{N} w_i \cdot p_i, 0 < w_i < 1, \tag{2}$$

where N represents the number of fields of the profile, p_i denotes the integrity of the ith field. Specifically, if the ith field is filled, p_i is 1, otherwise, p_i is 0. Considering the different contributions of different fields to detection, we set different weights for different fields, and w_i denotes the weight of the ith field. β_{CP} denotes the completeness of profile, its value range is $\left\{\beta_{CP} \middle| 0 \le \beta_{CP} \le 1\right\}$.

(5) **Comprehensive Level:** User level is highly correlated with the user's online duration and login habits. Compared with normal users, social bots usually have a shorter online duration and more irregular login habits in Sina Weibo, so they tend to have low user levels. Moreover, Sina Weibo has a function of official verification, and most verified users are normal users. Therefore, we define the comprehensive level as β_{CL}, it is calculated by

$$\beta_{CL} = \sum_{i=1}^{M} \iota_i \cdot u_i, 0 < \iota_i < 1, \tag{3}$$

where β_{CL} is the user's comprehensive level, ι_i is the value of the ith level, u_i is the weight of the ith level, and M is the number of levels. In this paper, we take user level and whether verification as the basis for calculating the comprehensive level. That is, whether verified (verification is 1, otherwise 0) and the normalized user level are weighted to calculate a user's comprehensive level. The value range of β_{CL} is $\left\{\beta_{CL} \middle| 0 < \beta_{CL} \le 1\right\}$.

3.2.2. Interaction-based features

The posts of users can be commented, reposted, and liked by other users. These interactions often reflect the difference between normal users and social bots. Therefore, based on the user's interactive behavior, we have extracted five interaction-based features.

(1) **Mean of the number of comments, reposts, and likes:** In [41], the number of a user's posts that are reposted is used as a feature. It can quantify a user's interaction with others. Many posts of social bots are illogical, and have few likes, comments, or reposts. Thus, we represent the mean of the number of comments, the mean of the number of reposts, and the mean of the number of likes on all posts of a user as γ_{MNCP}, γ_{MNRP}, and γ_{MNLP}, respectively. They are computed by

$$\begin{aligned} \gamma_{MNCP} &= \frac{1}{K}\sum_{i=1}^{K} \xi_i \\ \gamma_{MNRP} &= \frac{1}{K}\sum_{i=1}^{K} o_i \\ \gamma_{MNLP} &= \frac{1}{K}\sum_{i=1}^{K} \iota_i, \end{aligned} \tag{4}$$

where, ξ_i, o_i, ι_i are the number of comments, reposts, and likes of the ith post of a user, K is the number of posts of the user.

(2) **Diversity of sources of posts:** The posts of users usually come with post source, such as computer, mobile, etc. Normal users' posts usually have different sources, while social

bots' posts tend to have very few sources. Therefore, we consider the diversity of sources of posts as a feature and use the Margalef diversity index to calculate the feature, γ_{DSP}, which is given by

$$\gamma_{DSP}=\frac{\tau-1}{\ln K}, \tag{5}$$

where, τ denotes the number of types of sources.

(3) **Repost ratio:** The repost ratio of a user is the ratio of the total number of reposted posts to the total number of posts [2,42]. In most cases, the posts of social bots are copied from other users or generated using probabilistic methods. Thus, we use the repost ratio as a feature to distinguish between social bots and normal users. It is represented as γ_{RR} and computed by

$$\gamma_{RR}=\frac{\nu}{K}, \tag{6}$$

where ν denotes the number of reposted posts. The value range of γ_{RR} is $\left\{\gamma_{RR}\middle|0\le\gamma_{RR}\le 1\right\}$.

3.2.3. Content-based features

The content of different posts of social bots is often relatively similar, and the writing habits are generally illogical. Hence, we propose the following thirteen content-based features to distinguish users.

(1) **Mean and variance of the number of mentions in posts:** In Sina Weibo, users use "@" to mention other users when posting. Fazil et al. [2] considered the number of mentions in posts to distinguish users. Similarly, we define the mean and variance of the number of mentions in posts as δ_{MNMP} and δ_{VNMP}, respectively, and they can be computed by

$$\begin{aligned}\delta_{MNMP}&=\frac{1}{K}\sum_{i=1}^{K}\eta_i\\ \delta_{VNMP}&=\frac{1}{K}\sum_{i=1}^{K}(\eta_i-\delta_{MNMP})^2,\end{aligned} \tag{7}$$

where η_i is the number of mentions in the ith post.

(2) **Mean and variance of the number of hashtags in posts:** "#" is used by users to participate in the discussion of topics while posting a post. The number of "#" is considered to distinguish users in [2] and [30]. We take the mean and variance of the "#" number of posts as two features (δ_{MNHP} and δ_{VNHP}). And they can be computed in the same way as δ_{MNMP} and δ_{VNMP}.

(3) **Mean and variance of the number of URLs in posts:** Many social bots add URLs in posts to redirect visitors to external web pages for advertising [30], monetization, etc. In [2], they proved that the number of URLs plays a very important role in judging the quality of users' posts. Thus, the mean and variance of the number of URLs in posts are taken as features and represented as δ_{MNUP} and δ_{VNUP}, which can be computed in the same way as δ_{MNMP} and δ_{VNMP}.

(4) **Variance of the number of words in posts:** The word counts of different posts of a social bot are usually similar [40]. In our work, the variance of the number of words in posts (δ_{VNWP}) is given by

$$\begin{aligned}\delta_{MNWP}&=\frac{1}{K}\sum_{i=1}^{K}\zeta_i\\ \delta_{VNWP}&=\frac{1}{K}\sum_{i=1}^{K}(\zeta_i-\delta_{MNWP})^2,\end{aligned} \tag{8}$$

where ζ_i means the number of words of the ith post, and δ_{MNWP} is the mean of the number of words in posts.

(5) **Mean and variance of the number of punctuation marks in posts:** The use of punctuation marks in posts reflects a user's writing habits. In the posts of social bots, the frequency of punctuation marks is often unreasonable. For this reason, the mean and variance of the number of punctuation marks in posts are used as features, which are represented by δ_{MNPMP} and δ_{VNPMP}, and they are calculated like δ_{MNMP} and δ_{VNMP}.

(6) **Mean and variance of the number of interjection in posts:** An interjection is a word or expression that occurs as an utterance on its own and expresses a spontaneous feeling or reaction, such as "oh", "ah", "o", "ha", etc. These words can reflect a user's writing style. Thus, the mean and variance of the number of interjections in posts (δ_{MNIP} and δ_{VNIP}) are used in our work. To compute them, we follow the method used in δ_{MNMP} and δ_{VNMP}.

(7) **Mean of the score of the sentiment of posts:** Features of sentiment are the features extracted through sentiment analysis of the posts [40,43]. We analyze the sentiment polarity of posts and represent the mean of the score of the sentiment of posts as δ_{MSSP}, which is given by

$$\delta_{MSSP}=\frac{1}{K}\sum_{i=1}^{K}\rho_i, \tag{9}$$

where ρ_i is the score of the sentiment of the ith post. The value range of ρ_i is $\left\{\rho_i\middle|0\le\rho_i\le 1\right\}$. and that of δ_{MSSP} is $\left\{\delta_{MSSP}\middle|0\le\delta_{MSSP}\le 1\right\}$.

(8) **Variance of the number of pictures in posts:** When posting, users can add pictures to make their posts richer in content. Many social bots have almost the same number of pictures in their posts, while the number of pictures for each post of a normal user is usually not similar. This distinction is of great importance for detecting social bots. In our work, the variance of the number of pictures between each post of the user is represented as δ_{VNPP}, and it is calculated like δ_{VNWP}.

3.2.4. Timing-based features

Timing-based features are extracted from the time of users' posts. In [30], the authors found that there is a difference between social bots and normal users in the time distribution of posts. Consequently, the series of the time intervals between each post of a user is defined as $\theta=[\chi_1,\chi_2,\ldots,\chi_{K-1}]$, where K still denotes the number of posts of the user. Then, we use the following six timing-based features to distinguish users.

(1) **Mean and variance of the time interval between posts:** Chen et al. [15] considered the regularity of the time of users' posts in their research, the variance of the post time was taken as a feature. In our work, the mean and variance of the time interval between posts are represented as φ_{MTIP} and φ_{VTIP}, and they are defined by

$$\begin{aligned}\varphi_{MTIP}&=\frac{1}{K-1}\sum_{i=1}^{K-1}\chi_i\\ \varphi_{VTIP}&=\frac{1}{K-1}\sum_{i=1}^{K-1}(\chi_i-\varphi_{MTIP})^2,\end{aligned} \tag{10}$$

where χ_i is the time interval of two consecutive posts.

(2) **Longest and shortest time interval between posts:** Many social bots do not post for a long time after posting a large number of posts in a short time. Therefore, the longest and shortest time intervals between posts of the user are used

to distinguish between social bots and normal users in our work. We sort the series of the time interval to get a new series, $\theta' = [\chi_1', \chi_2', \ldots, \chi_{K-1}'] (\chi_i' \leq \chi_{i+1}', 1 \leq i \leq K-1)$. We take the mean of the user's shortest μ time intervals as the shortest time interval (φ_{STIP}), and the mean of the longest μ time intervals as the longest time interval (φ_{LTIP}). They are computed by

$$\begin{aligned} \varphi_{STIP} &= \frac{1}{\mu}\sum_{i=1}^{\mu} \chi'_i \\ \varphi_{LTIP} &= \frac{1}{\mu}\sum_{i=K-\mu}^{K-1} \chi'_i, \end{aligned} \tag{11}$$

After analysis, when $\mu = 5$, this pair of features can distinguish users well.

(3) **Burstiness parameters of the time interval between posts:** In [17], the author distinguished between normal users and social bots by using the burstiness parameters of the time intervals between posts and achieved great results. The burstiness parameters of time interval between posts (φ_{BPTIP}) is defined by

$$\varphi_{BPTIP} = \frac{\varphi_{VTIP} - \varphi_{MTIP}}{\varphi_{VTIP} + \varphi_{MTIP}} + \varepsilon, \tag{12}$$

where φ_{VTIP} and φ_{MTIP} are defined before, ε denotes the displacement factor. The displacement factor is a variable to meet the requirement of making the value of φ_{BPTIP} non-negative. There are three special values for φ_{BPTIP}: $\varepsilon - 1$, ε, and $\varepsilon + 1$, which can be understood as a completely regular behavior, a completely Poisson behavior, and the most bursty behavior, respectively [17]. Generally, the values of burstiness parameters of social bots are close to $\varepsilon - 1$ and $\varepsilon + 1$.

(4) **Information entropy of the time interval between posts:** In [17], the Shannon entropy was applied to quantify the regularity of the posting time interval series of users in detecting social bots and it worked well. In our work, the duplicate values in the posting time interval series of a user θ are removed to obtain a new series, $\theta'' = [\chi_1'', \chi_2'', \ldots, \chi_l'']$, where $l \leq K-1$. Then, we represent the Shannon entropy of the series of the time intervals as φ_{IETIP}. The calculation equation is given by

$$\begin{aligned} p(\chi_i'') &= \frac{n(\chi_i'')}{K-1} \\ \varphi_{IETIP} &= -\sum_{i=1}^{l} p(\chi_i'') \cdot \log(p(\chi_i'')), \end{aligned} \tag{13}$$

where $p(\chi_i'')$ is the frequency at which χ_i'' appears in the series $p(\chi_i'')$, $n(\chi_i'')$ denotes the number of times χ_i'' appears in the series θ. The smaller the Shannon entropy of the series of the time intervals φ_{IETIP}, the greater the probability that the user is a social bot.

3.3. Active learning for expanding labeled data

When using the deep learning-based approach to detect social bots in OSNs, a large-scale dataset is generally required to train models to achieve good performance. However, it is difficult to construct a large-scale dataset due to the difficulty of obtaining effective user data in OSNs and the high cost of manually labeling samples. Therefore, to expand the scale of the labeled dataset, we propose an approach for labeled data expansion based on active learning [21].

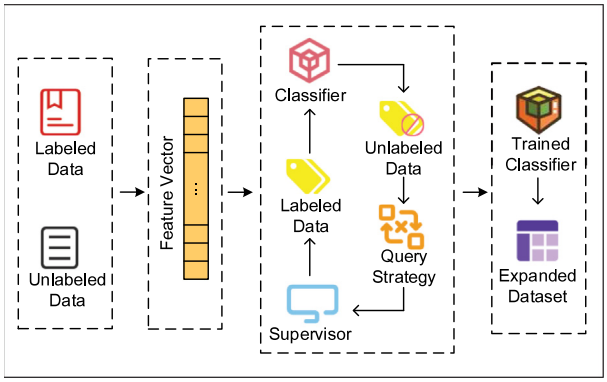

Fig. 2. The proposed labeled data expansion approach.

3.3.1. Expansion approach of labeled data

The proposed approach for labeled data expansion is shown in Fig. 2, and the working process of the approach is the following five steps:

(1) **Step 1:** Firstly, train an initial machine learning classifier by a small-scale labeled dataset. In the meantime, set the threshold for stopping training iterations.
(2) **Step 2:** Secondly, select a batch of the most valuable unlabeled user data according to the query strategy, and manually label them by the supervisor.
(3) **Step 3:** Thirdly, add these labeled user data into the labeled dataset, and update the parameters of the classifier using these data.
(4) **Step 4:** Fourthly, if the performance of the classifier exceeds the iteration stopping threshold, output the classifier, otherwise repeat step 2 and step 3.
(5) **Step 5:** Fifthly, predict the labels of the remaining unlabeled candidate samples and calculate the probability that they belong to the predicted labels using the classifier.
(6) **Step 6:** Finally, label the samples with high predicted probability, and conduct a sampling test after labeling.

3.3.2. Construction of a large-scale dataset

We have a labeled small-scale dataset $SWLD - 20K$ and an unlabeled dataset $SWUD$ containing potential normal users and potential social bots. Using the proposed approach for labeled data expansion, we further confirm the labels of potential normal users and potential social bots in $SWUD$ to construct a large-scale dataset. The construction process is described below.

To begin with, in the setting of the classifier, Mao et al. [44] used DT as the classifier in the study of active learning to expand the dataset, which can efficiently and accurately achieve large-scale expansion of the dataset. Therefore, we also choose the DT as the classifier for our active learning approach. Besides, we set the threshold for stopping training iterations: the accuracy of the classifier has not improved after ten consecutive training iterations. Meanwhile, according to the uncertainty sampling algorithm [45], we use an entropy-based uncertainty sampling algorithm as the query strategy, as shown in Algorithm 1. The query strategy is responsible for selecting a set S consisting of m samples with the largest entropy from the unlabeled dataset X. The calculation of the entropy for unlabeled samples is given by

$$e_i = -\sum_{j=0}^{1} P\left(y_j \,|x_i\right) \cdot \log P\left(y_j \,|x_i\right), \tag{14}$$

where, when j takes 0 and 1, $P\left(y_j \,|x_i\right)$ represents the probability that the unlabeled sample x_i is a normal user and a social bot,

Algorithm 1 Uncertainty query strategy based on entropy

Input:
An unlabeled dataset, $X = \{x_1, x_2, \cdots, x_n\}$, n denotes the number of samples in X;
A classifier that can ouput the classification probabilities, C;
The number of samples to select, m.

Output:
The most valuable sample set, S.

Procedure:
1: **for** i in n **do**
2: $P(y_1|x_i) = C(x_i)$;
3: $P(y_0|x_i) = 1 - P(y_1|x_i)$;
4: $e_i = -\sum_{j=0}^{1} P(y_j|x_i) \cdot \log P(y_j|x_i)$.
5: **end for**
6: Sort all e_i $(1 \le i \le n)$ to obtain E, $E = \{e_1, e_2, \cdots, e_n\}$;
7: Select top m samples based on E to form the set S.
8: **return** S.

respectively. Besides, e_i is the entropy value of the unlabeled sample x_i.

Specifically, when expanding labeled data, we first train the classifier using the dataset $SWLD - 20K$, and calculate the value of $P(y_j|x_i)$ for all potential normal users and potential social bots by the classifier. Subsequently, the entropy values for all the samples are computed and ranked. In each iteration, the 20 unlabeled samples with the largest entropy values are selected. After manually labeling these samples according to the proposed labeling metrics (see details in Section 3.1.3), we add them to the training dataset to retrain the classifier. When the conditions of stopping the iteration are satisfied, the iteration is stopped and we get the well-trained classifier. finally, the classifier is used to calculates the probability that a potential normal user is a real normal user and the probability that a potential social bot is a real social bot. For these unlabeled samples with a predicted probability value greater than 0.7500, their predicted label is taken as their true label. Then, these samples are added to the final dataset.

In our work, a total of 171 iterations are performed, a total of 3420 users are manually labeled according to the proposed labeling metrics, and finally, a classifier with the accuracy of 0.9801 is obtained. Using the classifier, the remaining potential normal users and potential social bots in *SWUD* are labeled. Mixing these newly labeled user data with user data in $SWLD-20K$ and further undertaking data balance processing, the dataset $SWLD - 300K$ (Sina Weibo labeled dataset with 300,000 samples) is built, the information of which is shown in Table 4. Moreover, a sampling test on the dataset $SWLD - 300K$ is undertaken with a sampling rate of 1%. To be specific, we randomly sample 1% of the data from the $SWLD - 300K$ dataset to test the correctness of the label. The final pass rate is 0.9910, which proves the effectiveness of the labeled samples.

3.4. The RGA model for detection

We design a novel deep neural network called RGA. The designed RGA model is mainly composed of a ResNet block, a BiGRU block, an attention layer, and an inference layer (see Fig. 3). The details of the model architecture are described as follows.

3.4.1. Model input

In the proposed framework, given a specific user, the user data is collected in the data collection and labeling module, and then features of the user are further extracted in the feature extraction module to form a feature vector, which can be denoted by $\mathcal{F} = \{f_1, f_2, \ldots, f_l\}$, and l is the number of features. To better suit the RGA model, we normalize the vector using the L_2 norm, without disrupting the linear relationship between the original data. Then the normalized feature vector of the user can be input into the RGA model.

Table 4
Overview of the $SWLD - 300K$ dataset.

Category	User number	Post number
Normal users	150,000	5,935,150
Social bots	150,000	4,833,979
Total	300,000	10,779,129

3.4.2. ResNet block

The research of Ismail Fawaz et al. [46] shows that ResNet has better performance in several deep learning models for time series classification. In our work, a feature vector of a user can be regarded as a time series. Therefore, ResNet is employed to extract temporal patterns of the feature vectors initially. In the designed RGA model, the ResNet block is mainly composed of three residual blocks. Each residual block is a multi-layer neural network containing convolutional 1D (Conv1D) layers, batch normalization (BN) layers, and rectified linear unit (ReLU) activation layers. Meanwhile, the input and output of each residual block can be directly connected through a shortcut connection. Therefore, a residual block can be defined as (taking the first residual block as an example)

$$\begin{aligned} h_1 &= \text{Convolution}_1(\mathcal{F}) \\ h_2 &= \text{Convolution}_2(h_1) \\ h_3 &= \text{Convolution}_3(h_2) \\ h_4 &= h_3 + \mathcal{F} \\ h' &= \text{ReLU}(h_4), \end{aligned} \tag{15}$$

where Convolution_i, $i \in \{1, 2, 3\}$ denotes the combination of a Conv1D layer, a BN layer, and a ReLU layer. h_1, h_2, h_3, h_4 represent hidden vectors. Moreover, h' is the output of the first residual block, which is transferred to the next residual block. The output vector of the last residual block is defined as $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$, which is as input of the BiGRU block.

3.4.3. BiGRU block

BiGRU is a kind of recurrent neural network (RNN) [23]. Since both the past state and the future state are considered, BiGRU can extract temporal patterns from data and has a better stability. Therefore, it is used to further extract the temporal patterns of feature vectors in our work. In the RGA model, the BiGRU block is composed of a forward GRU and a backward GRU. In each moment i $(i \in \{1, 2, \ldots, l\})$, the output vector of BiGRU is determined by the two one-way GRUs, it can be represented by

$$\begin{aligned} \overrightarrow{h_i} &= GRU_{fwd}\left(c_i, \overrightarrow{h_{i-1}}\right) \\ \overleftarrow{h_i} &= GRU_{bwd}\left(c_i, \overleftarrow{h_{i+1}}\right) \\ \overleftrightarrow{h_i} &= \overrightarrow{h_i} + \overleftarrow{h_i}, \end{aligned} \tag{16}$$

where $\overrightarrow{h_i}$ is the hidden vector of the forward GRU in time step i, and $\overleftarrow{h_i}$ is the hidden vector of the backward GRU in time step i. Moreover, the dropout technology is employed in the BiGRU block to suppress overfitting, so that the output of BiGRU block in each time step can be defined as h'_i.

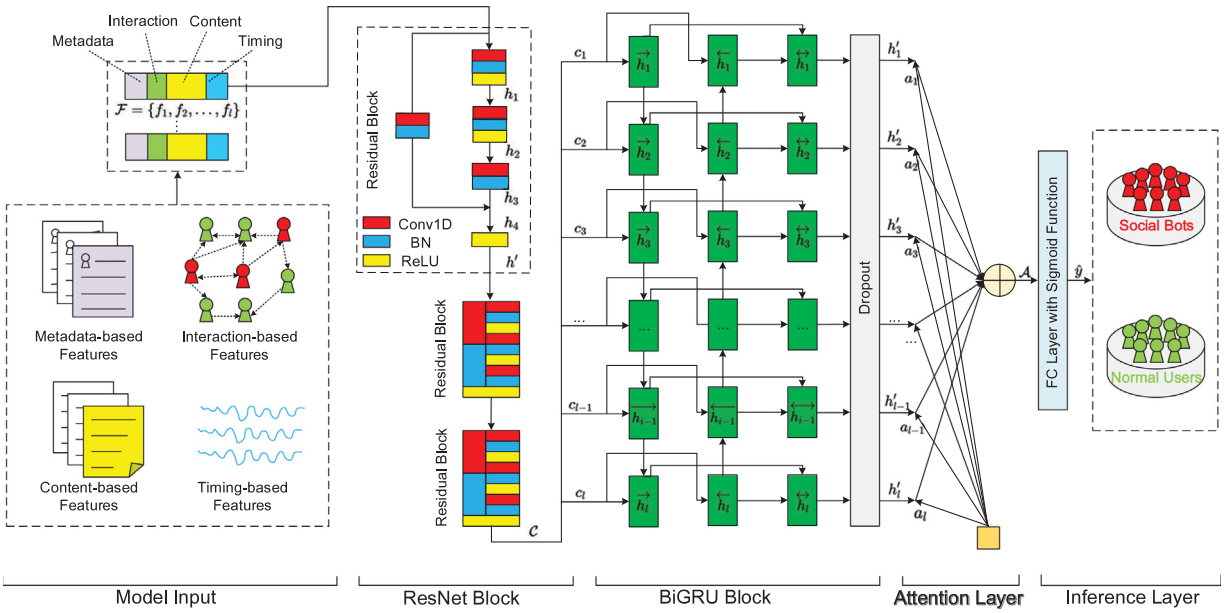

Fig. 3. The model architecture of the RGA deep neural network.

3.4.4. Attention layer

The attention mechanism is mainly to focus limited attention resources on key information [24]. We take advantage of the attention mechanism for that it can efficiently extract important patterns of sparse data to find useful information in the data that is significantly related to the current output, thereby improving the quality of output data. We define the importance weight of h'_i ($i \in \{1, 2, \ldots, l\}$) as a_i, the output of the attention layer can be denoted by

$$\mathcal{A} = \sum_{i=0}^{l} \alpha_i h'_i. \tag{17}$$

3.4.5. Inference layer

In the inference layer, a fully connected layer with a sigmoid activation function is employed to perform binary classification and output the classification results. The vector $\mathcal{A}$, calculated as the feature representation of the user, can be used to compute the probability that the user is a social bot by

$$\hat{y} = Sigmoid(\mathcal{A}). \tag{18}$$

The training loss function is defined by the negative log-likelihood of the correct labels, which can be computed by

$$\mathfrak{L} = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})), \tag{19}$$

where y is the true label of the user, that is, if the user is a social bot, the value of y is 1, otherwise, it is 0.

4. Experiments and evaluation

In this section, we evaluate the performance of the proposed DABot framework for social bot detection. To begin with, we describe the experiment settings in our work. Then, we evaluate the effectiveness of the features used in our work. Moreover, we conduct experiments to verify the superiority of the RGA model to detect social bots. Furthermore, the influence of the scale of the dataset on detection performance is also explored.

Table 5
The hyperparameters during deep learning model training.

Configuration	Value
Optimization function	Adam
Epoch	100
Batch size	32
Learning rate	0.001
ReduceLROnPlateau	monitor='val_acc', factor=0.5, patience=10, epsilon=0.0001
EarlyStopping	monitor='val_acc', patience=20, mode='max'
ModelCheckpoint	monitor='val_acc', mode='max', save_best_only=True

4.1. Experiment settings

Before describing the experimental design and results, the experiment settings are elaborated, including environmental setup, baseline studies, and performance metrics. The details of the experiment settings are as below.

4.1.1. Environmental setup

In our work, all experiments are undertaken on a workstation with an Intel Xeon E5-2618L v3 CPU and NVIDIA GeForce RTX 2080TI GPU with 64 GB of RAM. Each experiment is repeated ten times independently, and the average results are shown. All the classical machine learning models are built by the Scikit-learn library[1] Meanwhile, all the deep learning models are implemented using the Keras library[2] with the Tensorflow backend,[3] the configuration of the hyperparameters during deep learning model training is shown in Table 5.

4.1.2. Baseline studies

To conduct performance evaluation, a series of baseline studies are considered in our work. These state-of-the-art approach for detecting social bots are as follows:

[1] Scikit-learn: Simple and efficient tools for predictive data analysis (https://scikit-learn.org/).

[2] Keras: The Python deep learning library (https://keras.io/).

[3] Tensorflow: The end-to-end open-source machine learning platform (https://www.tensorflow.org/).

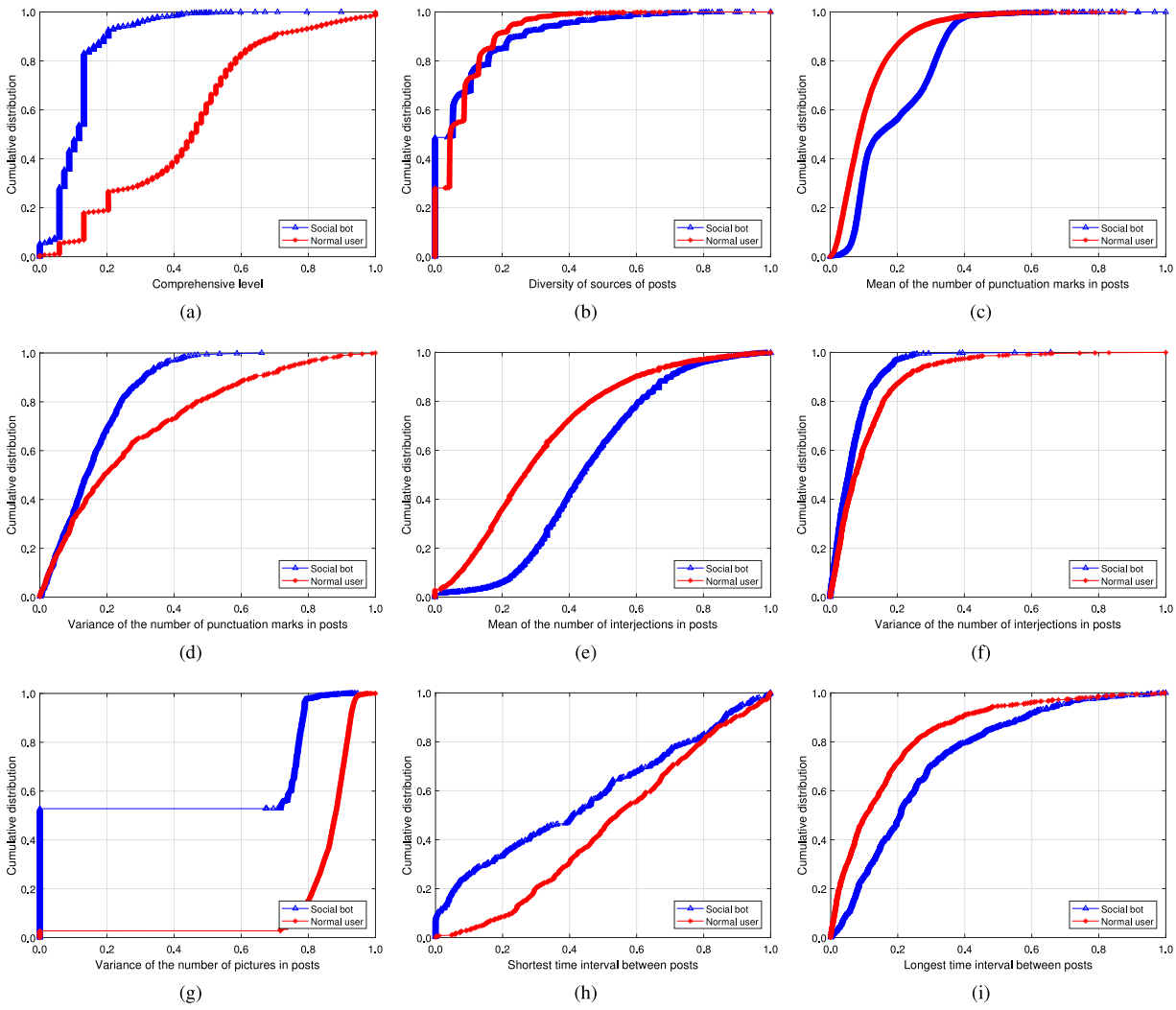

Fig. 4. Cumulative distribution plots of the proposed new features.

- **LR:** Logistic regression (LR) is a simple and powerful approach to solve some linear binary classification problems. It is widely used in the detection of social bots in Sina Weibo [47].
- **SVM:** SVM is another great approach of machine learning that has appreciable effects in detecting social bots in Sina Weibo [16].
- **ELM:** ELM was applied in [19] as an approach for social bot detection. ELM is considered to have advantages in learning rate and generalization ability.
- **RF:** Due to its good classification performance, scalability, and ease of use, RF has an excellent performance in the detection of social bots in [20].
- **MLP:** Multilayer perceptron (MLP), also referred to back propagation neural network (BPNN) in [42]. It is a feedforward neural network trained through the back propagation algorithm of error.
- **LSTM:** LSTM is one of the RNN, which helps in classification and regression problems. It is employed in detecting social bots and performs well [48].
- **ComNN:** ComNN referred to combined neural network proposed in [49] in our paper. ComNN includes an LR and two artificial neural networks (ANN) to incorporate different features and perform detection.

Table 6
Confusion matrix in social bot detection.

True	Predicted	
	Social bot	Normal user
Social bot	True positive	False negative
Normal user	False positive	True negative

- **CNN:** Convolution neural network (CNN) is one of the most widely used artificial neural networks, it is also effective in detecting social bots [50].

4.1.3. Performance metrics

A variety of metrics are used to evaluate the performance of the detection approaches including accuracy, recall, precision, and F1-score. The confusion matrix is employed to introduce these metrics, as shown in Table 6. The true positive (TP) is the number of social bots that are correctly detected, the false negative (FN) is the number of social bots that are incorrectly detected, the false positive (FP) is the number of normal users that are incorrectly detected, and the true negative (TN) is the number of normal users that are correctly detected. Then, accuracy, recall, precision, and F1-score can be computed by

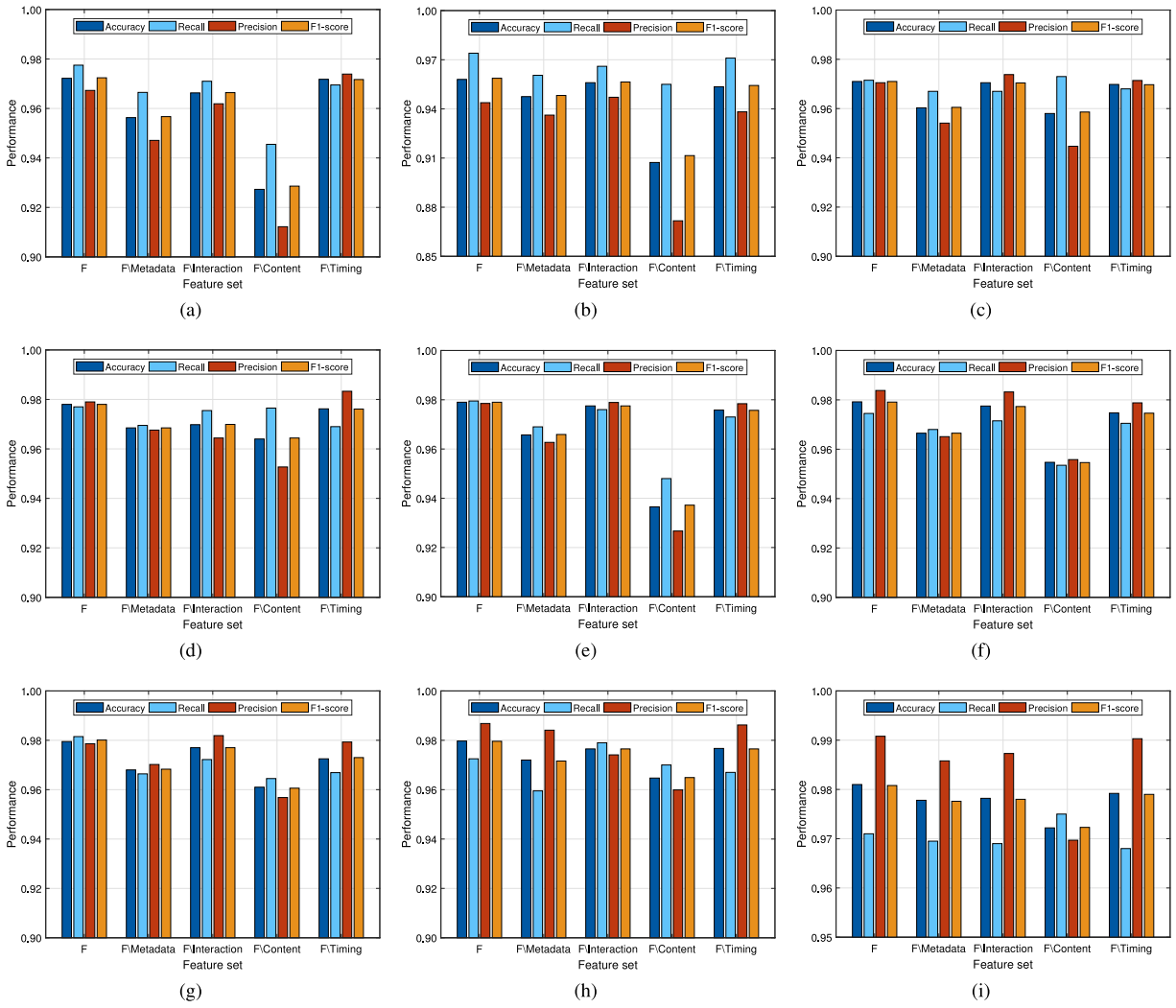

Fig. 5. The performance of the proposed RGA and baseline detection approaches in the feature ablation tests. (a) LR. (b) SVM. (c) ELM. (d) RF. (e) MLP. (f) LSTM. (g) ComNN. (h) CNN. (i) RGA.

$$\text{accuracy} = \frac{|\text{TP} + \text{TN}|}{|\text{TP} + \text{TN} + \text{FP} + \text{FN}|}, \tag{20}$$

$$\text{recall} = \frac{|\text{TP}|}{|\text{TP} + \text{FN}|}, \tag{21}$$

$$\text{precision} = \frac{|\text{TP}|}{|\text{TP} + \text{FP}|}, \tag{22}$$

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{23}$$

4.2. Evaluation of the effectiveness of features

We extract four categories of features: metadata-based, interaction-based, content-based, and timing-based features. In order to evaluate the validity of the proposed features before expanding the dataset, we conduct experiments on the features with the $SWLD - 20K$ dataset. Specifically, a statistical analysis of the new features is made to verify the discriminative power of these new features. Furthermore, we conduct feature ablation tests. That is, a category of feature is removed from the feature set each time, and then various detection approaches are used for testing to explore the contribution of each category of feature to detection.

Table 7
The description of feature sets.

Feature set	Categories of features included
F	Metadata, Interaction, Content, Timing
$F \backslash Metadata$	Interaction, Content, Timing
$F \backslash Interaction$	Metadata, Content, Timing
$F \backslash Content$	Metadata, Interaction, Timing
$F \backslash Timing$	Metadata, Interaction, Content

4.2.1. Discriminative power of new features

To evaluate the discriminative power of the proposed new features, we normalize the values of new features for all users and make cumulative distribution plots of them, as shown in Fig. 4. As indicated in the figure, the new features proposed in this paper significantly show great differences between social bots and normal users. Among these new features, the features of the comprehensive level, the diversity of sources of posts, the mean of the number of interjections in posts, the variance of the number of pictures in posts, and the minimum time interval between posts are more distinguishable. For instance, in terms of the comprehensive level, the values of nearly 90% of social bots are less than 0.2, and that of 80% of normal users are higher than 0.2. As for the diversity of sources of posts, almost 90% of social

Table 8
The architecture of different deep neural networks in model ablation tests.

Model	#Layers	#Conv1D	#BiGRU	Attention	Normalize	Activate	Regularize
ResNet (R)	11	9	0	–	Batch	ReLU	None
BiGRU (G)	2	0	1	–	None	Tanh	Dropout
ResNet-BiGRU (RG)	11	9	1	–	Batch	ReLU, Tanh	Dropout
ResNet-Attention (RA)	11	9	0	√	Batch	ReLU, Tanh	None
BiGRU-Attention (GA)	3	0	1	√	None	Tanh	Dropout
RGA	12	9	1	√	Batch	ReLU, Tanh	Dropout

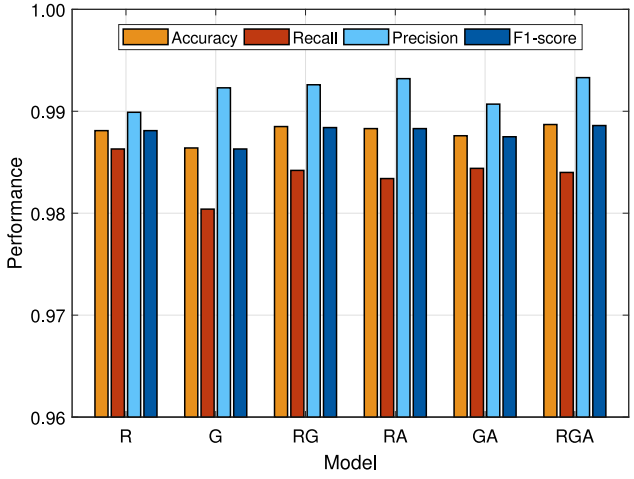

Fig. 6. Performance comparison of the different deep neural networks and proposed RGA.

bots have values that are smaller than 0.1, and 50% of normal users have values greater than 0.1.

4.2.2. Feature ablation tests

To evaluate the contribution of each category of feature to the detection performance, we perform feature ablation tests based on the full feature set and four subsets of the feature set. The subsets of the feature set can be represented by the set-difference function given as

$$F \backslash F' = \left\{ x | x \in F \wedge x \notin F' \right\}, \tag{24}$$

where F is the set with all features, F' is the subset of F with a particular category of features, and x is all user data of a feature. Table 7 shows the details of feature sets we used in feature ablation tests.

In this experiment, the $SWLD-20K$ dataset is divided into 60% for training, 20% for validation, and 20% for testing. Moreover, in addition to using the RGA detection approach, other baseline approaches are also employed to perform feature ablation study. The results are shown in Fig. 5. The performance of each approach with F, $F\backslash Metadata$, $F\backslash Interaction$, $F\backslash Content$ and $F\backslash Timing$ are compared. We find that all detection approaches perform best on the feature set F that contains all the features than on other feature sets. This proves that each category of extracted features in this paper has the distinguishability between social bots and normal users. In addition, all approaches perform worst using the feature set of $F\backslash Content$, which indicates that the distinguishability of content-based features is the greatest. Whereas the performance of approaches using feature set of $F\backslash Timing$ is similar to that of F, which indicates that the distinguishability of timing-based features is the smallest. We can also find that although the recall of RGA is not all the highest compared with the other five approaches on the same feature set, the accuracy, the precision, and the F1-score of RGA are the highest. That is, the RGA model has better detection performance than other

Table 9
Numerical results of the different deep neural networks and proposed RGA in detecting social bots.

Model	Accuracy	Recall	Precision	F1-score
ResNet (R)	0.9881	0.9863	0.9899	0.9881
BiGRU (G)	0.9864	0.9804	0.9923	0.9863
ResNet-BiGRU (RG)	0.9885	0.9842	0.9926	0.9884
ResNet-Attention (RA)	0.9883	0.9834	0.9932	0.9883
BiGRU-Attention (GA)	0.9876	0.9844	0.9907	0.9875
RGA	**0.9887**	**0.9840**	**0.9933**	**0.9886**

detection approaches on these feature sets. It is noted that the performance of the approaches still has much space to improve, so the dataset needs to be expanded to improve the detection performance.

4.3. Evaluation of the proposed detection approach

In order to evaluate the performance of the proposed RGA detection approach, the detection model ablation tests are launched, then the performance comparison with the baseline detection approaches is carried out. It is worth noting that these experiments are based on the $SWLD-300K$ dataset.

4.3.1. Detection model ablation tests

To verify that the deep neural network RGA proposed in this paper has certain advantages, we perform ablation tests the proposed model using $SWLD-300K$ dataset to provide an understanding of the impact of each layer or block of our model and show how significantly they affect the model performance. Specifically, the attention layer, BiGRU block, and ResNet are sequentially removed from the model, then the performance of the reduced model is evaluated. Furthermore, the combination of the ResNet block and attention layer and the combination of the BiGRU block and attention layer are sequentially removed to verify the validity of the proposed detection model. The architecture of the six deep neural networks in the experiment is shown in Table 8, which includes the number of layers, the number of Conv1D layers, the number of BiGRU layers, the use of the attention layer, the normalization strategy, the activation function, and the regularization strategy. In this experiment, the $SWLD-300K$ dataset is divided into 80% for training, 10% for validation, and 10% for testing, respectively. This is because the scale of the dataset is very large, 10% of data in the dataset is sufficient for the evaluation of the validation set or test set. In addition, more data in the training set can make deep neural networks more adequately trained.

Fig. 6 and Table 9 show the experimental results about the performance of several deep learning models, in terms of accuracy, recall, precision, and F1-score. We can see that the proposed RGA model has the best performance among these models on almost all metrics, with accuracy, recall, precision, and F1-score of 0.9887, 0.9840, 0.9933, and 0.9886 respectively. Meanwhile, each block or layer plays a role in the effectiveness of the RGA model, and the ablation of any block or layer can weaken the effect of the model. It is also obvious that the ResNet block contributes the

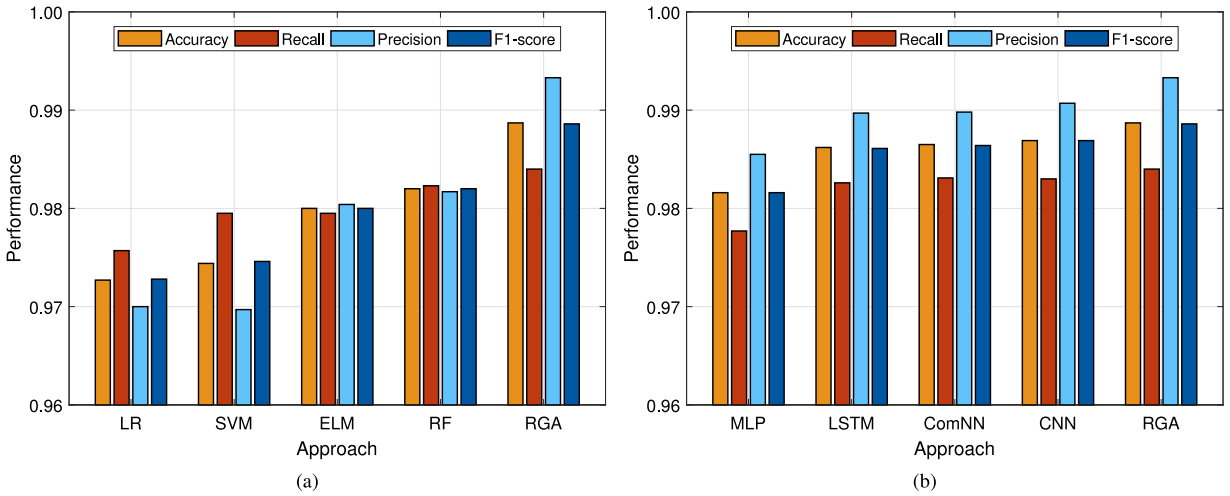

Fig. 7. Performance comparison of the baseline approaches and proposed RGA. (a) Performance comparison with classical machine learning approaches. (b) Performance comparison with deep learning approaches.

Table 10
Numerical results of the baseline approaches and proposed RGA in detecting social bots.

Approach	Accuracy	Recall	Precision	F1-score
LR [47]	0.9727	0.9757	0.9700	0.9728
SVM [16]	0.9744	0.9795	0.9697	0.9746
ELM [19]	0.9800	0.9795	0.9804	0.9800
RF [20]	0.9820	0.9823	0.9817	0.9820
MLP [42]	0.9816	0.9777	0.9855	0.9816
LSTM [48]	0.9862	0.9826	0.9897	0.9861
ComNN [49]	0.9865	0.9831	0.9898	0.9864
CNN [50]	0.9869	0.9830	0.9907	0.9869
RGA	**0.9887**	**0.9840**	**0.9933**	**0.9886**

most to the performance of the proposed RGA model, while the effect of the attention layer is relatively less obvious. Importantly, the proposed RGA model has generally the best performance over other reduced deep learning models in detecting social bots.

4.3.2. Performance comparison with the baseline approaches

To further verify the effectiveness of the proposed RGA model for detection, we compare the performance of RGA and the baseline detection approaches including four classical machine learning approaches (LR, SVM, ELM, RF) and four deep learning approaches (MLP, LSTM, ComNN, CNN) on the dataset $SWLD-300K$, which is split into 80% for training, 10% for validation, and 10% for testing, respectively.

Fig. 7 and Table 10 show the experimental results. It can be seen that on such a large dataset, deep learning approaches have certain advantages in performance over classical machine learning. Among the classical machine learning approaches, RF performs the best and LR performs poorly. In the meantime, CNN has better performance than other baseline deep learning approaches. Importantly, RGA exhibits the best performance in terms of accuracy, recall, precision, and F1-score, which proves that RGA has great advantages over the state-of-the-art approaches in detecting social bots.

4.4. Influence of the scale of the dataset

In our work, the dataset $SWLD-20K$ is constructed by manually labeling, and then the dataset $SWLD-300K$ is built by the proposed active learning approach for labeled data expansion. In order to verify the effectiveness of the proposed data expansion approach, we compare the performance of the RGA and baseline approaches on the two datasets, and the F1-score of these approaches on the two datasets is shown in Fig. 8.

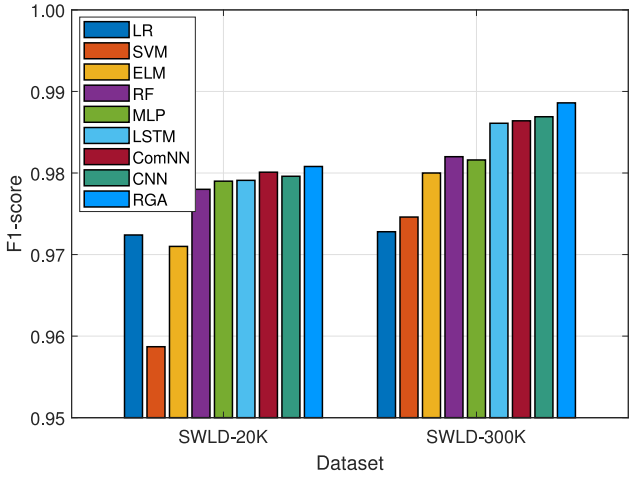

Fig. 8. Comparison the F1-score of the baseline approaches and proposed RGA using the two datasets.

It is obvious that the F1-score of all approaches is improved when active learning is used to expand the scale of the dataset from 20,000 to 300,000. It is worth noting that, on the whole, compared with classical machine learning approaches, the F1-score of the deep learning approaches has a greater increase. The exception is that the performance of the SVM is greatly improved, and the performance of the MLP is not significantly improved. Overall, the detection performance of these detection approaches can be improved using the proposed active learning approach for labeled data expansion.

Importantly, the proposed RGA has a significant improvement when the scale of the dataset becomes larger. Meanwhile, the proposed RGA not only performs best in the dataset with 20,000 samples but also has the best performance in the dataset with 300,000 samples. It is obvious that the proposed RGA detection approach is applicable to both small-scale and large-scale datasets.

5. Conclusion

This paper has proposed a novel DABot framework for detecting social bots with deep neural networks and active learning. The framework DABot mainly comprises four modules: data collection and labeling module, feature extraction module, active learning module, and detection module. Specifically, we extracted 30 features including nine completely new features of four categories, namely metadata-based, interaction-based, content-based, and timing-based features to achieve a comprehensive description of social bots in Sina Weibo. Moreover, this paper proposed an approach for expanding labeled datasets using active learning, which can significantly increase the efficiency of labeling user data and obtain a large-scale dataset at a small cost. Furthermore, making use of the ResNet, the BiGRU, and the attention mechanism, this paper designed a new deep neural network model RGA for detection. The experimental results showed that the proposed DABot framework is efficient for detecting social bots, and it has the best performance compared to other used detection approaches.

In OSNs, the attributes and behavior patterns of social bots continue to evolve to avoid detection. Therefore, making the detection approach adaptable to the evolution of social bots will be done in our further research work. In the meantime, there are clusters of social bots, that is, social bots tend to follow each other and form network clusters. The research on social bot cluster detection is also our future work.

CRediT authorship contribution statement

Yuhao Wu: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Yuzhou Fang:** Data curation, Methodology. **Shuaikang Shang:** Investigation, Visualization. **Jing Jin:** Investigation, Resources. **Lai Wei:** Validation, Visualization. **Haizhou Wang:** Funding acquisition, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61802270, 61802271, 81602935, and 81773548. The authors thank anonymous reviewers for their helpful comments to improve the paper.

References

[1] H. Liang, C.C. Li, G. Jiang, Y. Dong, Preference evolution model based on Wechat-like interactions, Knowl.-Based Syst. 185 (2019) 104998.
[2] M. Fazil, M. Abulaish, A hybrid approach for detecting automated spammers in Twitter, IEEE Trans. Inf. Forensics Secur. 13 (11) (2018) 2707–2719.
[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, A. Flammini, The rise of social bots, Commun. ACM 59 (7) (2016) 96–104.
[4] Z. Guo, Y. Shen, A.K. Bashir, M. Imran, N. Kumar, D. Zhang, K. Yu, Robust spammer detection using collaborative neural network in Internet of thing applications, IEEE Internet Things J. (2020) http://dx.doi.org/10.1109/JIOT.2020.3003802.
[5] M. Chakraborty, S. Pal, R. Pramanik, C. Ravindranath Chowdary, Recent developments in social spam detection and combating techniques: A survey, Inf. Process. Manag. 52 (6) (2016) 1053–1073.
[6] S.M. Alzanin, A.M. Azmi, Rumor detection in Arabic tweets using semi-supervised and unsupervised expectation–maximization, Knowl.-Based Syst. 185 (2019) 104945.
[7] G. Lingam, R.R. Rout, D.V. Somayajulu, Adaptive deep Q-learning model for detecting social bots and influential users in online social networks, Appl. Intell. 49 (11) (2019) 3947–3964.
[8] C. Shao, G.L. Ciampaglia, O. Varol, K.C. Yang, A. Flammini, F. Menczer, The spread of low-credibility content by social bots, Nature Commun. 9 (1) (2018) 1–9.
[9] M. Al-Qurishi, M.S. Hossain, M. Alrubaian, S.M.M. Rahman, A. Alamri, Leveraging analysis of user behavior to identify malicious activities in large-scale social networks, IEEE Trans. Ind. Inform. 14 (2) (2017) 799–813.
[10] S. Kudugunta, E. Ferrara, Deep neural networks for bot detection, Inform. Sci. 467 (2018) 312–322.
[11] H. Ping, S. Qin, A social bots detection model based on deep learning algorithm, in: Proceedings of the 18th IEEE International Conference on Communication Technology, 2018, pp. 1435–1439.
[12] Q. Cao, X. Yang, J. Yu, C. Palow, Uncovering large groups of active malicious accounts in online social networks, in: Proceedings of the 21st ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 477–488.
[13] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, K. Beznosov, H. Halawa, Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs, Comput. Secur. 61 (2016) 142–168.
[14] G.C. Santia, M.I. Mujib, J.R. Williams, Detecting social bots on Facebook in an information veracity context, in: Proceedings of the 13th International AAAI Conference on Web and Social Media, 2019, pp. 463–472.
[15] H. Chen, J. Liu, Y. Lv, M.H. Li, M. Liu, Q. Zheng, Semi-supervised clue fusion for spammer detection in Sina Weibo, Inf. Fusion 44 (2018) 22–32.
[16] Q. Fu, B. Feng, D. Guo, Q. Li, Combating the evolving spammers in online social networks, Comput. Secur. 72 (2018) 60–73.
[17] J. Pan, Y. Liu, X. Liu, H. Hu, Discriminating bot accounts based solely on temporal features of microblog behavior, Phys. A Stat. Mech. Appl. 450 (2016) 193–204.
[18] F. Wu, J. Shu, Y. Huang, Z. Yuan, Co-detecting social spammers and spam messages in microblogging via exploiting social contexts, Neurocomputing 201 (2016) 51–65.
[19] X. Zheng, X. Zhang, Y. Yu, T. Kechadi, C. Rong, ELM-based spammer detection in social networks, J. Supercomput. 72 (8) (2016) 2991–3005.
[20] H. Fu, X. Xie, Y. Rui, Leveraging careful microblog users for spammer detection, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 419–429.
[21] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, H. Zha, Joint active learning with feature selection via CUR matrix decomposition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (6) (2019) 1382–1396.
[22] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, in: Proceedings of the 30th International Joint Conference on Neural Networks, 2017, pp. 1578–1585.
[23] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1422–1432.
[24] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.
[25] M. Latah, Detection of malicious social bots: A survey and a refined taxonomy, Expert Syst. Appl. 151 (2020) 113383.
[26] N.Z. Gong, M. Frank, P. Mittal, SybilBelief: A semi-supervised learning approach for structure-based sybil detection, IEEE Trans. Inf. Forensics Secur. 9 (6) (2014) 976–987.
[27] Z. Yang, J. Xue, X. Yang, X. Wang, Y. Dai, VoteTrust: Leveraging friend invitation graph to defend against social network sybils, IEEE Trans. Dependable Secure Comput. 13 (4) (2015) 488–501.
[28] B. Wang, L. Zhang, N.Z. Gong, SybilSCAR: Sybil detection in online social networks via local rule based propagation, in: Proceedings of the 36th IEEE International Conference on Computer Communications, 2017, pp. 1–9.
[29] X. Zhang, H. Xie, J.C. Lui, Sybil detection in social-activity networks: Modeling, algorithms and evaluations, in: Proceedings of the 26th IEEE International Conference on Network Protocols, 2018, pp. 44–54.
[30] Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of Twitter accounts: Are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. 9 (6) (2012) 811–824.
[31] C. Yang, R. Harkreader, G. Gu, Empirical evaluation and new design for fighting evolving Twitter spammers, IEEE Trans. Inf. Forensics Secur. 8 (8) (2013) 1280–1293.
[32] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, A.H. Wang, Twitter spammer detection using data stream clustering, Inform. Sci. 260 (2014) 64–73.
[33] C. Cai, L. Li, D. Zeng, Detecting social bots by jointly modeling deep behavior and content information, in: Proceedings of the 26th ACM Conference on Information and Knowledge Management, 2017, pp. 1995–1998.
[34] S. Wen, W. Liu, Y. Yang, P. Zhou, Z. Guo, Z. Yan, Y. Chen, T. Huang, Multi-label image classification via feature/label co-projection, IEEE Trans. Syst. Man Cybern. Syst. (2020) http://dx.doi.org/10.1109/TSMC.2020.2967071.

[35] J. Lu, J. Xuan, G. Zhang, X. Luo, Structural property-aware multilayer network embedding for latent factor analysis, Pattern Recognit. 76 (2018) 228 – 241.
[36] N. Chavoshi, H. Hamooni, A. Mueen, DeBot: Twitter bot detection via warped correlation, in: Proceedings of the 16th IEEE International Conference on Data Mining, 2016, pp. 817–822.
[37] S. Cresci, R.D. Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Social fingerprinting: Detection of spambot groups through DNA-inspired behavioral modeling, IEEE Trans. Dependable Secure Comput. 15 (4) (2017) 561–576.
[38] T. Zhao, M. Malir, M. Jiang, Actionable objective optimization for suspicious behavior detection on large bipartite graphs, in: Proceedings of the 6th IEEE International Conference on Big Data, 2018, pp. 1248–1257.
[39] M. Khayat, M. Karimzadeh, J. Zhao, D.S. Ebert, VASSL: A visual analytics toolkit for social spambot labeling, IEEE Trans. Vis. Comput. Graphics 26 (1) (2019) 874–883.
[40] O. Varol, E. Ferrara, C.A. Davis, F. Menczer, A. Flammini, Online human-bot interactions: Detection, estimation, and characterization, in: Proceedings of the 11th International AAAI Conference on Web and Social Media, 2017, pp. 280–289.
[41] S. Mohammad, M.U. Khan, M. Ali, L. Liu, M. Shardlow, R. Nawaz, Bot detection using a single post on social media, in: Proceedings of the 3rd World Conference on Smart Trends in Systems, Security and Sustainability, 2019, pp. 215–220.
[42] Y. Lian, X. Dong, Y. Chi, X. Tang, Y. Liu, An Internet water army detection supernetwork model, IEEE Access 7 (2019) 55108–55120.
[43] O. Loyola-González, A. López-Cuevas, M.A. Medina-Pérez, B. Camiña, J.E. Ramírez-Márquez, R. Monroy, Fusing pattern discovery and visual analytics approaches in tweet propagation, Inf. Fusion 46 (2019) 91–101.
[44] L. Ma, S. Destercke, Y. Wang, Online active learning of decision trees with evidential data, Pattern Recognit. 52 (2016) 33–45.
[45] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3–12.
[46] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.A. Muller, Deep learning for time series classification: A review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963.
[47] L. Liu, K. Jia, Detecting spam in Chinese microblogs - A study on Sina Weibo, in: Proceedings of the 8th International Conference on Computational Intelligence and Security, 2012, pp. 578–581.
[48] A. Makkar, N. Kumar, An efficient deep learning-based scheme for web spam detection in IoT environment, Future Gener. Comput. Syst. 108 (2020) 467 – 487.
[49] W. Pei, Y. Xie, G. Tang, Spammer detection via combined neural network, in: Proceedings of the 14th International Conference on Machine Learning and Data Mining in Pattern Recognition, 2018, pp. 350–364.
[50] Z. Alom, B. Carminati, E. Ferrari, A deep learning model for Twitter spam detection, Online Soc. Netw. Media. 18 (2020) 100079.